On Gamma Ray Instrument On-Board Data Processing Real-Time Computational Algorithm for Cosmic Ray Rejection

Semion Kizhner, Stanley D. Hunter, Andrei R. Hanu and Teresa B. Sheets
National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt Road, Greenbelt MD, 20771
301-286-1294
Semion.Kizhner-1@nasa.gov

Abstract—**Richard O. Duda and Peter E. Hart of Stanford Research Institute in [1] described the recurring problem in computer image processing as the detection of straight lines in digitized images. The problem is to detect the presence of groups of collinear or almost collinear figure points. It is clear that the problem can be solved to any desired degree of accuracy by testing the lines formed by all pairs of points. However, the computation required for n=NxM points image is approximately proportional to n^2 or $O(n^2)$, becoming prohibitive for large images or when data processing cadence time is in milliseconds. Rosenfeld in [2] described an ingenious method due to Hough [3] for replacing the original problem of finding collinear points by a mathematically equivalent problem of finding concurrent lines. This method involves transforming each of the figure points into a straight line in a parameter space. Hough chose to use the familiar slope-intercept parameters, and thus his parameter space was the two-dimensional slope-intercept plane. A parallel Hough transform running on multi-core processors was elaborated in [4]. There are many other proposed methods of solving a similar problem, such as sampling-up-the-ramp algorithm (SUTR) [5] and algorithms involving artificial swarm intelligence techniques [6]. However, all state-of-the-art algorithms lack in real time performance. Namely, they are slow for large images that require performance cadence of a few dozens of milliseconds (50ms). This problem arises in spaceflight applications such as near real-time analysis of gamma ray measurements contaminated by overwhelming amount of traces of cosmic rays (CR). Future spaceflight instruments such as the Advanced Energetic Pair Telescope instrument (AdEPT) [7-9] for cosmos gamma ray survey employ large detector readout planes registering multitudes of cosmic ray interference events and sparse science gamma ray event traces' projections. The AdEPT science of interest is in the gamma ray events and the problem is to detect and reject the much more voluminous cosmic ray projections, so that the remaining science data can be telemetered to the ground over the constrained communication link. The state-of-the-art in cosmic rays detection and rejection does not provide an adequate computational solution. This paper presents a novel approach to the AdEPT on-board data processing burdened with the CR detection top pole bottleneck problem. This paper is introducing the data processing object, demonstrates object segmentation and distribution for processing among many processing elements (PEs) and presents solution algorithm for the processing bottleneck – the CR-Algorithm. The algorithm is based on the a priori knowledge that a CR pierces the entire instrument pressure vessel. This phenomenon is also the basis for a straightforward CR simulator, allowing the CR-Algorithm performance testing. Parallel processing of the readout image's (2(N+M) – 4) peripheral voxels is detecting all CRs, resulting in O(n) computational complexity. This algorithm near real-time performance is making AdEPT class spaceflight instruments feasible.**

TABLE OF CONTENTS

1. INTRODUCTION

The AdEPT instrument prototype development is in an early development phase. So there is, naturally, *no expectation* of having the instrument's full implementation *design details*, including optimized on-board data processing algorithms. However, there are already quite a few known and invariant in time general **conceptual** properties (requirements that are not expected to change) of the **A**dEPT instrument and the instrument's **O**n-Board Data **P**rocessing **C**omputational **A**lgorithms (**OPCA**). The cosmic ray detection and rejection or **CR-Algorithm** is the most computationally complex part of the OPCA (**the top pole bottleneck**). Finding a method to reduce the CR-Algorithm's computational complexity and to **parallelize the CR-Algorithm** goes a long way in removing the OPCA from the list of the AdEPT instrument development **top poles**. These concepts are delineated below, followed by the **outline** of the **parallelizable and real time Cosmic Ray detection CR-Algorithm** that is based on these concepts. *The purpose of this paper is to demonstrate that the CR-Algorithm can be segmented and parallelized. It depicts* as well that the CR-Algorithm computational complexity is reduced from

non-linear **O(n^2)** to linear **O(n)** so that it can ran on an *inexpensive* distributive *computational system and be executed in real-time*.

2. CONCEPTS OF THE ON-BOARD DATA PROCESSING

The AdEPT instrument **T**ime **P**rojection **C**hamber (**TPC**) is a 2mx2mx2m cube (or cylinder) comprising the upper and lower half chambers filled with a mixture of two pressurized gasses. The instrument TPC concept is depicted below and in **Figure 1** as a 2mx2mx2m cube (or right 2m diameter and height cylinder) vessel divided in the middle by a plain cathode (maintained at -100KV) and with two readout planes of detectors on the top and bottom faces of the cube. The two detector readout planes are presented in color brown and the high voltage cathode plane is shown in color green. The TPC and the science measurements are described in detail in **Figure 2**.

The AdEPT TPC and micro-well detector (MWD) concepts were developed at NASA by Stanley Hunter [7].

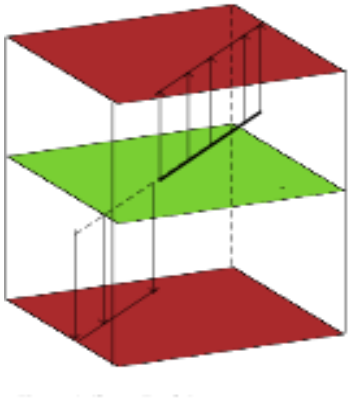


Figure 1. Charge Track Projection onto the upper and lower sensor planes

The above 2m x 2m x 2m TPC concept in Figure 1 and Figure 6 depicts the CR trace in gas and its corresponding color projections in the two detector readout planes (color brown). The two readout planes contain a total of 5Kx10K MWDs or voxels [7] that can't be read out with state-of-the-art readout technologies. Instead, rows and columns of MWDs of the non-bonded 50cm x 50cm x 50cm modules are read out (total of *81024 channels)*. These fulfill the science requirements for the instrument resolution and full sky gamma ray detection and its polarization study. After the channel readout and digitization the data is used to update the 5Kx10K image processing object on-board using the CR-algorithm first to detect and reject the CRs to reduce the volume of downlink telemetry. The TPC internal vertical surface is armored with a *charge drift cage* directing the gamma ray and CR gas interaction charges to drift vertically at a selected velocity to one of the detector readout planes depending on the ion emergence location in the TPC. The ions drift to a readout plane filled with multitudes (5Kx5K) of micro-well detectors whose charge readout, digitization and on-board data processing allows to obtain the event of interest vector near-linear projections (line traces). The readout at 20μs cadence allows reconstructing the event vector origin vertical axis component. The ion vertical drift in the electrical field is assured by maintaining the TPC content at a thermal equilibrium. The volume of information derived from the detector readout planes after digitization to 16-bits is $8x10^8$ bits that need to be transmitted to the ground in 20μs at $4x10^{13}$ bits per second. This exceeds the state-of-the-art communication width of 150MHz by an order of 10^5. The gamma rays related traces constitute a fraction of a percent and its information can be transmitted to the ground. The on-board data processing goal is to detect and reject the interference CR events' traces that constitute the bulk of traces in the detector readout planes leaving only gamma ray traces for the telemetry stream. It is impractical to readout all the detector voxels. Instead rows and columns are read out with the columns maintained at +300V and row at -300V bias with the signal being accumulated on the negative electrodes and the nearly equal induced charge on the positive electrodes Y-axis (**Figure 3 and Figure 4**).

The related invariant concepts are delineated as follows:

The AdEPT readout plane contains approximately 5Kx5K detectors. There are two readout planes comprising an **n**=5Kx10K voxels conceptual readout image. It is known that a relativistic energy CR enters the instrument pressure vessel and exits the vessel, resulting in a trace that extends from the readout plane image peripheral point of entry to the peripheral point of exit. This presents the concept of processing only the image peripheral points instead of each of the point pairs in the image, resulting in computational complexity reduction from the **O(Cxn^2) ~ O(10^{17})** to **O(Cxn) ~ O(10^{10}).** The number of operations performed on a voxel is accepted as **C=10**.

Because of the very large number of the AdEPT instrument detector voxels (**5Kx10K**) and large number of their read-out channels (**81,024** of rows and columns with read out at **20μs** cadence) the on-board data processing computational complexity is estimated as **O(10^{17})** floating-point operations per second (**FLOPS**). This requires on-board **super-computing capabilities** and a **new parallelizable linear computational complexity algorithm as stated above, O(10^{10}).**

The AdEPT on-board data processing of the above complexity of **O(10^{10})** floating-point operations per second (FLOPS) is further reduced to 50 sub-algorithms each of complexity **O($2x10^8$).** The algorithm of such complexity can be handled by 200MHz processing

elements. We observe that there are ((2x5K + 2x10K) – 4) or approximately 30K of peripheral points in the AdEPT **5Kx10K** readout image. The upper boundary for computational complexity on the n=30K peripheral voxels is then Cxn^2 = 30Kx30K and C=10 resulting in $O(10^{10})$. Distributing processing among 50 PEs results in $O(10^{10} / 50)$ or $O(2x10^8)$ computational complexity of a sub-algorithm and can be executed on 200MHz processing elements. This computation can be achieved by a distributed computational platform comprising an electronics card carrying a computer chip and a set of around 50 programmable logic devices (PLDs) or processing elements (PEs), such as FPGAs, DSPs, GPUs. The OPCA itself or its bottleneck algorithms such as the CR-Algorithm must be susceptible to parallelization as described below.

The AdEPT on-board processing must be susceptible to segmentation and **parallelization**, so that smaller input data segments can be processed on a set of processing elements at 20Hz cadence. For this the input data streams must be organized into a "**processing object**" that can be **segmented** and **processed in parallel** on a distributive computational platform in 50ms or at 20Hz cadence. Not all processes can be parallelized. For example, **processing a streaming input of data is almost impossible to segment and parallelize.**

The OPCA processing cycle cadence is **50ms or 20Hz.** The 50ms stems from the time required for an **anion** to drift the longest vertical path in the AdEPT gas vessel of 1 meter drift distance from the vessel mid-cathode to one of the two readout planes.

We can't reset the instrument vessel gas to "neutral" or a state of no trace charges at the beginning of each 50ms data processing cycle, even if we to switch off the drift cathode -100KV plate. *However, remaining projection's charges in motion from previous processing cycle (not yet reached the readout planes' voxels), after the previous cycle image is cleared, will be read out at the next processing 50ms cycle.*

AdEPT instrument comprises two readout planes each of approximate size of 5K x 5K and when conceptually "stitched" they forming an image of size **5K x 10K** to be processed in 50ms. **This image is the processing object**" that can be **segmented** and **processed in parallel** on a distributive computational platform in 50ms or at 20Hz cadence (**Figure 2**).

The OPCA in present conceptual view comprises 5 modules or Algorithms. These algorithms are elaborated upon in the following Section by their enumeration and functionality.

3. ON-BOARD ALGORITHMS ENUMERATION

There are four data conditioning algorithms preceding the five computational algorithms.

The AdEPT on-board computational process comprises the following five algorithms:

OPCA={PO-Algorithm, SD-Algorithm, NR-Algorithm, CR-Algorithm, GR- Algorithm}

- **Algorithm 5 (PO)** for input data stream(s) arrangement into a 50ms duration **computational processing object** susceptible to segmentation
- **Algorithm 6 (SD)** Segmentation and Distribution of the computational processing object among multiple cores for parallel processing using Algorithms 7-9
- **Algorithm 7 (NR)** Noise features removal
- **Algorithm 8 (CR)** Cosmic Ray detection and removal (**priority CR-Algorithm**)
- **Algorithm 9 (GR)** Gamma-Ray event extraction and download to SC**.**

The algorithms are running on a distributed computational platform comprising a **single CPU** and as many as **50 200MHz** processing elements (**Figure 2**).

3.1 Algorithms Characterization

The algorithms must be susceptible to "parallelization". Namely they should be capable of being divided into smaller pieces and running on many processing elements, say processing **50 1Kx1K** sub-images of the large 5Kx10K image in 50ms, each on a one of 50 processing elements. As already described above - ***processing a streaming input of data is almost impossible to segment and parallelize.***

3.1.1 Algorithm considerations

Each algorithm requires formulation, prototype, IV&V implementation, and firmware considerations. These are further delineated in Section 3.1.2.

Formulation - algorithm science formulation by the AdEPT science team, constituting the **Algorithm Theoretical Basis** in a form of a text document. It can also be viewed as the Algorithm's **rationale** which must provide the algorithm computational complexity in the number of floating-point operations per second (FLOPS), say **$O(10^{11})$.**

Prototype - development of the science computational algorithm prototype in some high-level language, say MATLAB that has **the toolbox** for data image segmentation and segment processing assignment to different cores on a multi-core CPU, such as the MAC 3.5 GHz **6-Core** Intel Xeon E5 processor.

IV&V Implementation - algorithm prototype initial implementation in C++ to Independently Verify that the algorithm solves the stated problem and that Validate that it works (IV&V).

Firmware - the implementation of the C++ code bottlenecks will then be further designed in the selected computational platform programmable Logic Devices (PLD) firmware using C++ code as design specifications for the firmware design. The PLD are used to realize the computational platform super-computing capabilities.

3.1.2 Algorithm required considerations details

To facilitate the algorithm development in C++ verification input and corresponding output data set must be simulated in MATLAB and saved in the same data set as the algorithm prototype

The **processing object** must be susceptible to graphic display, For example processing object as an image allows input data image being displayed by MATLAB functions.

The simulated input must be simply basic so that it could be well understood and replicated. It must be in the form of detector plane channels readout and output of the X-axis and Y-axis data channels from the Concentrator to the computational platform CPU

Readout planes image elements that are detected for rejections are to be filled with MATLAB **NaN code - 999999.99.** Any **feature** in an image is comprised of a set of points, say {F}. These filled with NaNs can be cleared (removed) by a simple scan over an image point-by-point and looking for Nans and zeroing them out.

The gas vessel may contain multitudes of CR traces during a 50ms processing cycle but **t**he probability of two traces intersecting in the readout planes is negligible.

However, intersection of trace projections is abundant. Still, intersection of a few projections at the same point is also very small, say 5.

The stitched image can be realized as an array of single precision floating point real numbers with 4-bytes for the measurement time stamp and 4 bytes for the measurement amplitude: Real I(5K, 10K, 5, 2).

This requires on-board random access memory (**RAM**) in the amount of (5K x 10 K x 5 x 2) x 4 bytes or 5K x 10K x 5 x 8 = 2000 * K *K = 2,000, 000, 000 = **2GB** of **RAM**. With a buffer this amounts to **4GB RAM requirement.**

The CR-Algorithm is the OPCA process bottleneck because there is so much more CR events in the Gas vessel compared to Gamma Ray events, like “a needle in a stack of hay”. Constructing the CR-Algorithm is among the AdEPT tall poles.

3.2. ON-BOARD PROCESSING FUNDAMENTALS

The AdEPT On-Board Processing comprises Computational Object Formation, Object Segmentation and Distribution, CR- Algorithm Cases 1-4. The following **Figure 2** depicts the two 5Kx5K readout plains conceptually stitched as a single image of size 5Kx10K. This single image represents the processing object accumulated during the 50ms detector planes readout at the 50Hz cadence. At the end of the 50ms cycle the processing object is formed and segmented into 50 1Kx 1K sub-images that are distributed among 50 PEs for processing using the CR-Algorithm.

3.3 On-Board Processing Object Formation, Segmentation and Distribution

The AdEPT instrument is comprised of two chambers separated by a cathode plane (C- color yellow) depicted below in **Figure 2**. The two 5Kx5K detector readout planes (RP1 and RP2 – in green) are correspondently the upper and lower readout planes. *These can be visualized as one large matrix of size NxM where N=10K and M=5K.*

The computational object formation, segmentation and distribution for computation among the many processing elements is depicted below in **Figure 2**.

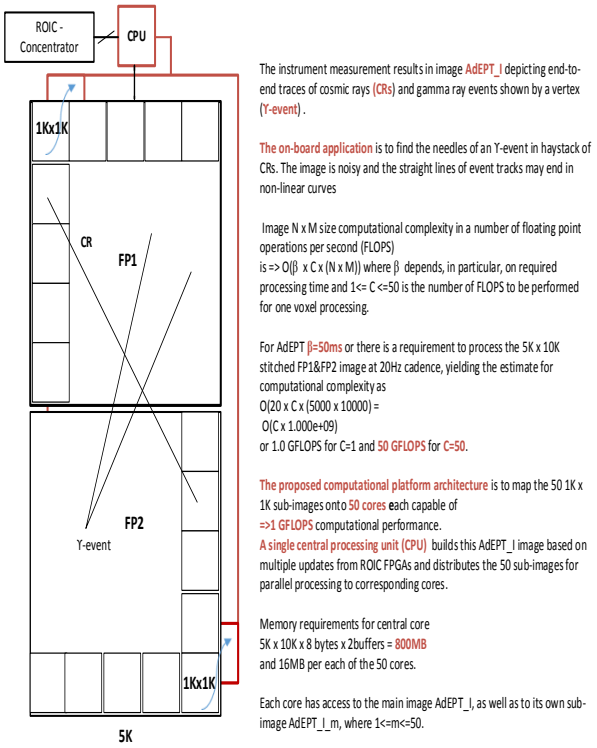


Figure 1. AdEPT two 5K x 5K focal planes FP1 and FP2 stitched as a single 5K x 10K image AdEPT_I and symbolically divided into 50 1K x 1K sub images AdEPT_I_m, where 1<=m<=50.

Figure 2. Two readout planes stitched as one matrix.

3.4 Processing Data Object Formation

The processing object is the stitched image depicted in **Figure 2**. It is initialized at 50ms processing cadence and then updated each 20μs readout for 50ms or 2500 times using the X-axis and Y-axis readouts of the detector planes. The X-axes are charge-event readouts and the corresponding Y-axes carry almost equal induced charges. The problem is that an event charged voxel on some readout X_i and Y_j may not be singular charged voxels (**Figure 3**), so that the event can be located by comparing for equal amplitudes the X_i and Y_j readouts. These axes readouts can carry additional charged voxels, say when an X_i and an Y_j are carrying 4 voxels in total on each axis (Figure 4).

Example 1. Search for two X-axis and Y-axis at readout time t=t1 such that

$$\mathbf{a}(X_i) = \mathbf{a}(Y_j) \text{ yielding } \mathbf{A_{ij} = a}$$

Example 2.

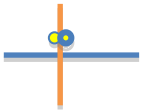


Figure 3. Typical readout.

Search for X-axis and Y-axis readouts pairs (x_{i1}, x_{i2}), (y_{j1}, y_{j2}) such that the system of 4 linear equations in 4 unknowns is solvable

$$\mathbf{(a}(x_{ik}) + \mathbf{a}(x_{il})) + (\mathbf{a}(x_{jk}) + \mathbf{a}(x_{jl})) = (\mathbf{X_i} + \mathbf{X_j})$$
$$\mathbf{(a}(y_{ik}) + \mathbf{a}(y_{il})) + (\mathbf{a}(y_{jk}) + \mathbf{a}(y_{jl})) = (\mathbf{Y_k} + Y_l)$$
$$\mathbf{a}(x_{ik}) + \mathbf{a}(x_{il}) = \mathbf{X_i}$$
$$\mathbf{a}(x_{jk}) + \mathbf{a}(x_{jl}) = X_j$$
where $(\mathbf{X_i} + \mathbf{X_j}) = (\mathbf{Y_k} + Y_l)$

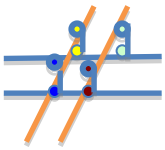


Figure 4. Detector Rows (blue) and Columns (brown) readout lines.

Searching for pairs of X-readout pair and Y-readout pair that satisfy equality (circles of same color depict induced charges)

$(X_i + X_j) = (Y_k + Y_l)$ requires C_K^2 x C_K^2 operations which is doable on a distributed computational system as $O(K^2 \times K^2)$. Readout at t1:

Xi(ai, t1) 1<=i<=40,000 (when modules are all bonded)
Yj(aj,t1) 1<=j<=40,000

Concentrator of 5% of non-zero amplitude readouts
Xk(ak,t1) 1<=k<=1K / 20 = 50, say X1, X5, X 17
Yl(al, t1) 1<=l<=1K / 20 = 50, say Y3, Y7, Y21

Corollary:
Number of non-zero Xs comprising voxels = Number of non-zero-induced Ys voxels that are comprising the Ys readout.

The ideal case is when there is one voxel comprising an Xi readout and one voxel comprising and Yj readout in concentrator 50x50 output:

Xk(ak,t1) 1<=k<=50, say X1, X5, X 17
Yl(al, t1) 1<=l<= 50, say Y3, Y7, Y21

At t1 we compare X1(a) with {Y3, Y7, Y21} to find the single matching amplitude Y(a), say the amplitude are comparable at Y7. Then (X1, Y7) have the same amplitude "a" and comprise the **image point x1,7(a, t1).** We then go on and repeat this process with X5 and X7.

This requires 50x50=2500 operations only for each of the (50ms=50,000 μs)/20μs = **2500 t-readouts**.

This is how we find the minimum set of x1p1, x5p2, x17p3 points in the image:

(X1, Yp1, X1(a), t1), (X5,Yp5, X5(a),t1), (X17, Yp17, X17(a), t1)

We could do it for 1<=t<=2500 readouts in the 50ms, however this would require an enormous amount of memory:

5Kx10Kx2500 = 5Kx10Kx2.5K x4bytes= 125KxKxKx4 = 125 x 10^9 x 4= 1000Gigabytes of RAM.

To address this issue we are going to store only a small time-readout subset, say 5 times instead of 2500. This will reduce the RAM requirement to 5Kx10Kx5x4bytes = 1000 x K x K = 10^9 = **1 GigaByte of RAM array, which is pretty acceptable**. This is only needed to reconstruct the Z-axis. We can now just abstract ourselves and reshape the image floating point 5Kx10Kx5 to image 5Kx10K and use it as the computational object for detecting projections of CR-events. This requires 5Kx10Kx4 = 200* 10^6 or 200MB of RAM.

The examined cases of 4 voxels constitute the majority cases of interest with other cases having very low probability based on empirical knowledge and being omitted from analyses. A few assumptions may help in constructing the image A. These are partially based on a short readout time of 20μs.

Assumption 1
Readout at time t_l of both axis comprise a single non-zero amplitude voxel then these must be equal (since one induced the other) and we compare all non-zero voxels of X_l all non-zero voxels of Y_l for matching amplitudes (5% of 1K each or 50x50 matrix search).

The 1K stems from the fact the 50cm^2 modules 1Kx1K readout anodes and cathodes (axes) are not bonded and the instrument readout axes are only 1K long each, as opposed to 4K if the module axis were bonded.

The AdEPT high-fidelity simulations produce output X and Y projection data streams that are governed by the above assumption.

Assumption 2
Subsampling within the 20μs can alleviate this issue to improve resolution along the readout axes.
I this paper we assume that the stitched image can be formed by the *targeted design* of the readout X-axes and Y-axes data streams.

3.5 Complexity reduction from $O(n^2)$ to O(n)

In an n=NxM matrix there are (2x(N+M) – 4) peripheral pixels. Indeed, adding up all peripheral pixels and subtracting four pixels counted twice we have (N + N + M + M - 4) =2x(N + M) -4 or approximately **2x(N + M)**.

We prove that the computational complexity of an algorithm based on analysis of peripheral points only is of linear complexity O(n), where n=NxM<=N^2 and

$n^2=(NxM)^2$ for N<M. Now, consider for N>M and n<N.

$(2x(N+M))^2 = 4(N+M)^2 = 4x(N^2 + 2NxM + M^2)$ <=

$4x(N^2 + 2NxN + N^2) = 4x(4xN^2) = 16N^2$

Since $N^2 \sim n$,

$$O(2x(N+M)) \sim O(16N^2) \sim O(n).$$

4. BOTTLENECK CR-ALGORITHM

4.1 Cases CR1-CR4

This section represents the Science Formulation and is the theoretical basis documentation of the CR-Algorithm. We are considering cases CR1, CR2, CR3 first, followed by case CR4.

The two readout planes RP1 and RP2 are interpreted as 5K x 5K arrays. Furthermore the arrays are stitched together, resulting in a 5K x 10K array. This array is extended by another "projection intersection" dimension of size 5 and storing voxel readout time and signal magnitude, resulting in an array of real numbers **I(5K, 10K, 5, 2)** being formed each 50ms or at 20Hz cadence. This array after being updated (filled) with data from the readouts in each 50ms is considered to be **t**he **OPCA processing object** to be processed in the next 50ms. This processing object is then segmented into **50 1Kx1K** sub-images and distributed among 50 cores for processing in parallel: **process a 1Kx1K image in 50ms** by **all the algorithms**, including this **CR-Algorithm**. This process is described in more detail in **Reference [7].**

The OPCA processing comprises reducing readout noise, detection and rejection of cosmic ray events and detecting gamma ray events for outputting to the spacecraft C&DH for telemetering to the ground station(s) for **science data processing**:

- The most intensive processing is to detect and reject the cosmic ray event traces since these constitute the bulk of projections in the two readout planes.

- Solving the CR-Algorithm **even approximately** will go a long way towards reducing the processing for the remaining traces.

- The algorithm computational complexity **lower boundary** was derived from interpreting the computational object as the 5Kx10Kx5x2 4-D array and multiplying all the dimensions to yield the number of voxels **n** and their processing cadence of 50ms or 20Hz and estimated number of floating-point operations on a single voxel as 10: 5K*10K*5*2 * 20Hz cadence * **C=10** FLOPS per voxel processing O(C*n), where C<<n

$$(n=(5*10*5*2*20*\ 10^{3}\ *\ 10^{3)}\ *\ (\mathbf{C=10})\ =$$
$$(5*5*2*2\ *\ 10*10*10)\ *\ 10^{6}\ =$$
$$100*10^{3}\ *\ 10^{6}\ =\ 10^{11}\ =\ \mathbf{O(10^{11})}$$

- The most intensive processing is to detect and reject the cosmic ray event traces since these constitute the bulk of projections in the two readout planes.

- Solving the CR-Algorithm **even approximately** will go a long way towards reducing the processing for the remaining traces.

The upper boundary for the computational complexity is $\mathbf{O(n^2)}$ or O((C=>n)*n), required to examine **all voxel pairs** for collinearity, yielding

$$(n=(5*10*5*2*10^{3}*10^{3}))\ *\ (C=>(5*10*5*2*10^{3}\ *\ 10^{3})\ )=\ 2500\ *\ 2500\ *\ 10^{12}\ =\ 6.25\ *\ 10^{6}\ 10^{12}\ =$$
$$\mathbf{O(10^{18})}$$

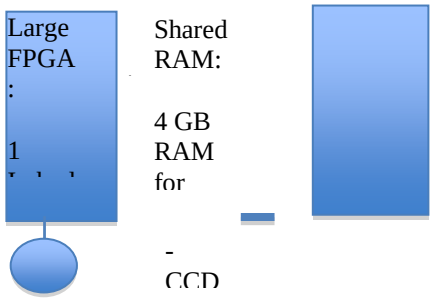
4.2 On Event Traces in Detector Readout Planes

Even accepting the empirical position the instrument readout planes voxels have 5% (or factor of 20 reduction) occupancy with non-zero amplitude events most of which are CRs and, in addition, the gamma ray events are statistically I per each CR event, we still have an enormous volume of data to process. Namely, the number of non-zero magnitude voxels n_0 can be estimated as

$$n_0 = (5K\ x\ 10K)\ /\ 20\ =2.5\ x\ 10^{6}\ \text{voxels}$$
$$(n_0)^2 = (Cx(2.5\ x\ 10^{6})\ ^{2})\ \sim 10^{14}$$

Furthermore, with gamma ray events to CR event ratio of 10, 000 and data volume reduction by an order of 4 magnitudes we still need to seek a better CR-Algorithm for the on-board processing to become feasible. Utilizing an inexpensive distributed computational system we can further reduce the computational complexity to $O(10^{8})$.

4.3 On-Board Data Processing Platform Conceptual Architecture



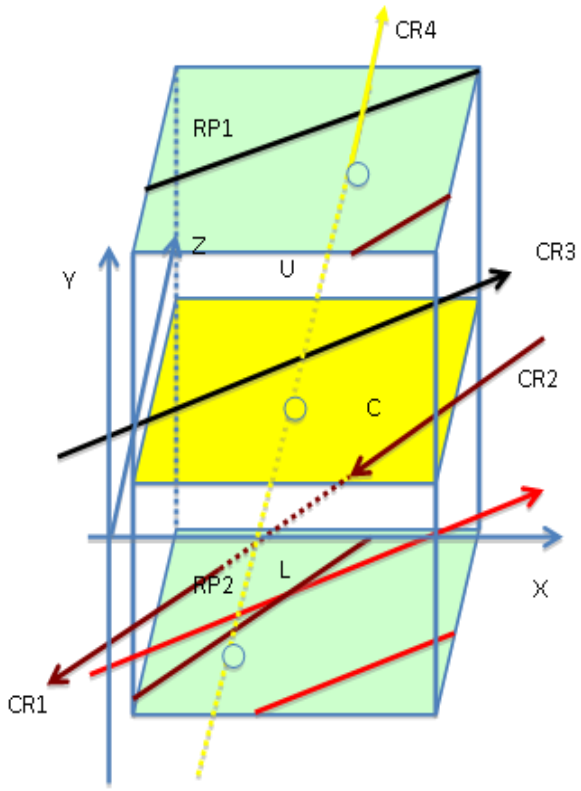


Figure 6. AdEPT Time Projection Chamber, comprising the instrument Upper (U) and Lower (L) half chambers.

The CR traces CR1-CR4 in color brown, red, black and yellow and their corresponding projections in detector readout planes R1, R2 – color green are depicted above in **Figure 6**.

Both lower and upper boundaries require unrealistic computational complexity and **reducing the lower boundary by a few orders of magnitude using an efficient algorithm, segmented processing object and distributive processing is the goal of this paper**. Namely, if we could restrict the CR trajectories search using mostly array boundary points this would reduce the computational complexity of CR detection to (5K + 5K + 5K + 5K) Upper readout plane periphery voxels + (5K + 5K + 5K + 5K) Lower readout plane periphery voxels = 40K or = $4*10^4$ voxels or $4*10^4$ * 20Hz * **100**FLOPS = **$O(10^8)$**

This estimate of boundary-based processing computational complexity **$O(10^8)$** has the potential of reducing the computational complexity lower boundary **$O(10^{11})$** by **3-orders of magnitude. This, together with the computational object segmentation and processing distribution among many PEs allows the CR-Algorithm to run on-board in real time.**

Detection of CR traces in a large 3-D volume from a large 5Kx10K readout trace projection planes RP1 & RP2 in a 50ms cadence is a known outstanding complexity problem [7].

Figure 6 components are as follows:

CR Case1 U&L
or CR1
CR Case2 L
or CR2
CR Case3 U
or CR3
CR Case4
L &C&U

We begin with considering **ideal image straight lines** (**Figure 6**)

The CR traces and associated projects are presented in the same color (**Figure 6).**

A CR trace cuts thru the gas vessel at entry and exit points.

As was indicated above, we can't reset the instrument vessel gas to "neutral" or no charges or traces at the beginning of each 50ms processing cycle, even if we switch off the drift cathode -100KV plate. However, remaining projection charges in motion from the previous processing cycle (not yet reached the readout planes' voxels), after the previous cycle image is cleared, will be read out at the next processing 50ms cycle. This was already described above but repeated here as the algorithm rationale.

In the readout planes RP1 and RP2 the CR trace projection segment(s) have at least one end on the plane boundary (a boundary or near –boundary point). This allows us to analyze in a 5Kx10K array just 5K+10K +5K+10K = 30K points instead of 5Kx10K points or 50KxK / 30K =1667K times less points or **three order of magnitude improvement in search of the CR projection initial points**.

There are **four** major CR-trace related cases: CR1, CR2, CR3, CR4. For cases CR1 and CR2 the trace projection is completely located in one of the readout planes RP1 or RP2.

For case CR3 when the trace crosses the vessel Upper and Lower half the corresponding projection comprises two segments – one in RP1 and the other in RP2 (**Figure 6**).

4.4 CR-Algorithm for Cases 1-3

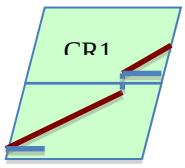
The view of the projection readout planes as a stitched 5K x 10K image demonstrates the CR-Algorithm ability to be **parallelized** by dividing the image into 50 1Kx1K images and running the CR-algorithm on 50 processing elements (**PEs**).

The CR1, CR2, CR3 are distinct from a γ-event comprising a vertex.

CR1 case is determined by boundary points and the same angle ***α***.

Once the CR ideal projection line is detected and marked off by NaNs the surrounding this line data points can be treated as noise and removed by marking them off by NaNs. **This is a very important point in the CR Algorithm**, as shown below in **Figure 7**.

Removing the CR remaining near-trajectory stand-alone points (**Figure 8**) can then be treated as a noise reduction process that is computationally simple.



This is the CR-Algorithm initial outline for Cases 1, 2, 3.

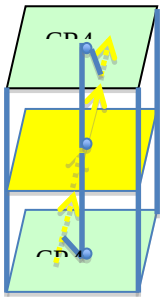
Figure 8. CR projection with adjacent artifact near collinear points that become "orphans" once the CR trajectory is detected and removed.

4.5 Conceptual Outline of the On-Board Processing's CR-Algorithm for Case CR4

CR4 event criterion.
Only a CR can cross the Cathode plane C. **The projection** of the CR4/Cathode intersection point with coordinates C0=(Cx_i, Cy_i) **on both readout planes** results in two voxels with the same coordinates in RP1 ($RP1x_i$, $RP1y_i$) and in RP2 ($RP2x_i$, $RP2y_i$) that are having a **non-zero amplitude**.

The probability of CR4 events is low. That is to say that even if we do not detect all CR4 events, the detected CR1, CR2 and CR3 events greatly reduce the remaining volume of work to be done.

The probability of two points with same coordinates in the two readout planes having non-zero amplitude is low. Only a CR4 event has such a characteristic. Together with the above CR4 event **criterion this provides a good characteristic of a CR4 event trajectory end point in RP1 and RP2**
.
Once the characteristic voxel in RP1 and RP2 corresponding to a CR4 event is detecting by a single sweep of the two readout planes RP1 and RP2 of size 5Kx5K each and checking amplitudes of the voxels with the same coordinates 1<=i <= 5Kx5K running on 50 PEs, we can determine the CR4 two projections emanating from point #i that satisfies the CR4 event criterion described above (**Figure 7** and **Figure 8**). and **Figure 9**.



Read out plane RP1 A($RP1x_i$, $RP1y_i$) > 0

Figure 9. Case CR4 of cosmic rays crossing all three AdEPT planes {L, C, U}}.

5. CR TRACE DETECTION IN READOUT DETECTOR PLANES

CR straight-line segment trajectory or trace determination is based on a high amplitude peripheral end point (small brown circle) neighborhood (larger green circle encompassing three adjacent points – in color green and blue) analysis, as follows:

Check CR end point neighborhood points for large amplitudes to determine the direction of the CR projection segment (blue dotted line) as shown below in **Figure 10**.

Once the end point (x1, y1) (small brown circle) and the nearest non-zero amplitude color blue point (x2, y2) are determined, the remaining points on the blue dotted line that is connecting them has the equation

$$y - y1 = m(x - x1), \text{ where } m = (y2 - y1) / (x2 - x1)$$

This line's points (**Figure 10**) may then be safely initialized to NaNs as the content of a CR projection.

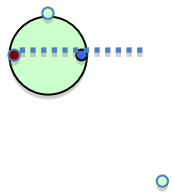


Figure 10. CR-Algorithm end-point neighborhood

6. CR-ALGORITHM PROTOTYPE OUTLINE AND CR EVENTS SIMULATOR

The above sections comprise the AdEPT on-board processing CR-Algorithm. With that done we can now proceed with CR-Algorithm prototyping in MATLAB (Trademark of MathWorks) language. Namely, we prototyped the algorithm cardinal modules [10]:
Processing Object - Prototyped input data **processing object** as the large array or real numbers
I(5ZK, 10K, 5, 2).

Parallelize Processing - Segmented processing **object image** I(5K, 10K,…) into 50 smaller sub-images
$\mathbf{I_i(1K, 1K,…)}$, where 1<=i<=50 and distribute the processing among 50 PEs in the distributive computational platform.

CR-Algorithm Prototype - For the CR-Algorithm Prototype implemented the few pages of MATLAB code using the above CR-Algorithm theoretical background sections as a specification.

Simulator – We coded a simple simulator to generate traces for CR1-CR-4 cases in a number N large enough to test algorithm prototype performance. We conducted few initial tests for N=50 with sufficient performance.

CONCLUSIONS

We presented the cosmic ray detection and rejection CR-Algorithm for gamma ray spaceflight instrument on-board data processing real-time computations. We have shown that the computational complexity of this class of algorithms is $O(n^2)$ and how to reduce it to O(n), that it can be segmented and parallelized. This allows it to run in real time on inexpensive computational platforms. In turn, this algorithm's real-time performance makes AdEPT class instruments feasible.

REFERENCES

[1] Richard O. Duda and Peter E.Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures", Communications of the ACM, January 1972, Vol. 15, #1
[2] Rosenfeld, A. "Picture Processing by Computer", Academic Press, New York, 1969
[3] Hough, P. V. C. Method and means for recognizing complex patterns - U.S. Patent 3,069,654, Dec. 18, 1962
[4] Chen, Y. K., Li, W., Li, J., and Wang, T., "Novel parallel Hough transform on multi-core processors" in Acoustics, Speech and Signal Processing. (ICASSP 2008), 1457{1460 (2008)
[5] J. D. Offenberg, D. J. Fixsen and J. C. Mather "Memory-Efficient Up-the-Ramp Processing with Cosmic-Ray Rejection" Received 2004 January 29; accepted 2004 November 18; published 2004 Dec. 29.
[6] Ulrich Kirchmaier, Simon Hawe and Klaus Diepold, "A line detection and description algorithm based on swarm intelligence", Technische Universitat Munchen, Arcisstrasse 21, 80290 Munich, Germany, 2009

[7] Stanley D. Hunter et al. "Medium Energy Gamma-Ray Astrophysics", White Paper Submission to Astro 2010: The Astronomy and Astrophysics Decadal Survey, February 15, 2009

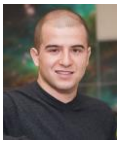
[8] Stanley D. Hunter et al. "A Pair Production Telescope for Medium-Energy Gamma-Ray Polarimetry" in Astroparticle Physics Volume 59, July–August 2014, Pages 18–28

[9] Andrei R. Hanu "AdEPT 50cm^2 Prototype Hardware System Architecture", 12-04-2015, NASA GSFC Report
[10] Semion Kizhner, AdEPT End-to-End Systems Engineering Narrative Report 2016, NASA GSFC Report

BIOGRAPHY

Semion Kizhner is an aerospace engineer with the National Aeronautics and Space Administration at the Goddard Space Flight Center. He proposed the development of the Hilbert-Huang Transform Data Processing System (HHT-DPS) for one dimension (1-D) and has been leading the HHT-DPS development team. He also proposed in 2013-2014 and as Principal Investigator developed the HHT2 system for 2-D. He participated recently in evaluation of the NASA Advanced Space Technology proposals, was recently the EE subsystem lead for the DESTINY/Joint Dark Energy Mission/WFIRST concept studies and Mars-2020 PING instrument proposal EE subsystem. He was the lead for the SMAP radiometer science data processing code development. He published three dozens of technical papers and mentored numerous undergraduate, graduate and doctoral students in the NASA Education Programs. He graduated from Johns Hopkins University with an MS degree in computer science and holds a Bachelors degree in applied mathematics and cybernetics.

Dr. Stanley D. Hunter is the AdEPT instrument Principal Investigator. Stanley Hunter is a senior astrophysicist in the NASA/GSFC Astroparticle Physics Laboratory. His gamma-ray astrophysics interests include study of the diffuse emission from the Galaxy, molecular clouds and supernova remnants. Since 1989 he has led the gamma-ray detector group and the effort to develop instrumentation for high angular resolution gamma-ray telescopes. This work has resulted in the development of the two-dimensional Micro-Well Detector (MWD) and the Three-Dimensional Track Imager (3-DTI). The 3-DTI is the enabling technology for the Advance Energetic Pair Telescope (AdEPT) a pair production telescope for medium energy gamma-ray polarimetry. He graduated from the University of Arizona with a B.S. in Physics, and B.S. in Mathematics. He received his Ph.D. in Physics from the Louisiana State University.



Andrei Hanu is a NASA Postdoctoral Fellow (NPP) in the NASA/GSFC Astroparticle Physics Laboratory. During his Ph.D. studies he developed an imaging detector using a THick Gaseous Electron Multiplier (THGEM) coupled to a two-dimensional delay line readout. His expertise includes development of gaseous imaging detectors, radiation spectrometry, Monte Carlo simulation techniques, dose estimation, and the development of high-speed analog/digital pulse processing systems. In December 2013 he joined the NASA/GSFC Astroparticle Branch and focused on the development of the Advanced Energetic Pair Telescope (AdEPT). He graduated from the McMaster University with a B.Sc. Medical & Health Physics. He received his Ph.D. Medical Physics from the McMaster University.

Teresa B. Sheets is the AdEPT on-board algorithms development lead. Teresa Sheets is a computer scientist with the National Aeronautics and Space Administration at the Goddard Space Flight Center Science Data Processing Branch. Ms. Sheets came to Goddard in 1980 as a Co-operative Education Student working with the Data Management and Programming Office of the Laboratory for High Energy Astrophysics. After graduating in 1982, she returned to the Data Management and Programming Office. She has supported many projects for scientists studying cosmic rays, x-rays, and gamma rays, providing support for telemetry and data processing, using Fortran, then C on Linux operating systems. She is familiar with many telemetry formats, including the CCSDS and LEO-T format standards. Supported projects include: Compton Telescope Balloon GSE, Alice Balloon GSE, Astro-1/BBXRT, GRO/BATSE, WIND/TGRS & KONUS, GCN, AdEPT, SpaceCube, and several IRADs. She graduated from the University of Tennessee with a Bachelors degree in Computer Science.